

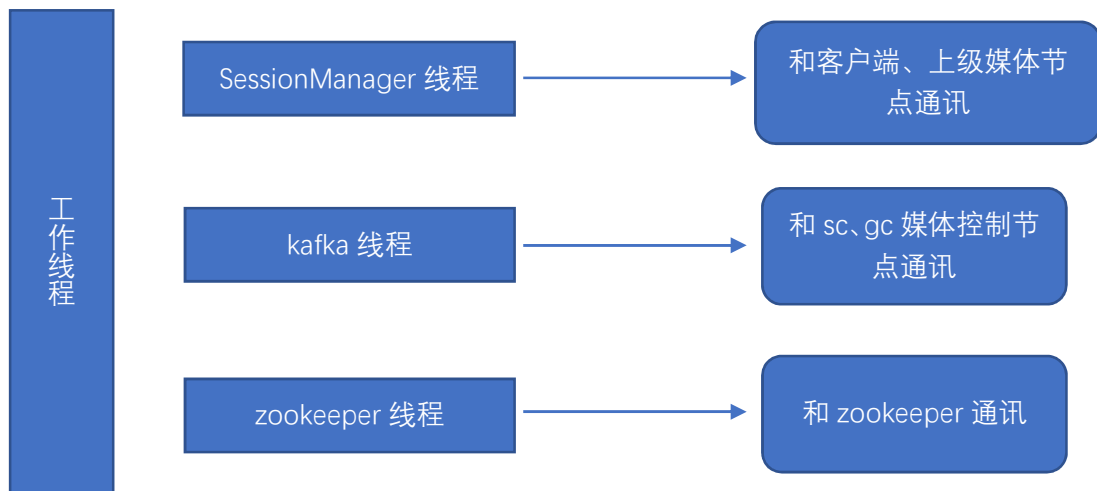
代码说明

一、背景说明

fsp 服务器媒体节点在面对高并发用户登录、高并发用户订阅流、高并发访问 redis、高并发和集群媒体控制节点通讯时，为了提高并发效率，采用多线程异步处理模型，以满足服务器应对高并发需求。

二、设计思路

此处就拿 fsp 媒体节点 gs 中的部分代码逻辑实现讲解，多线程模型架构图如下：



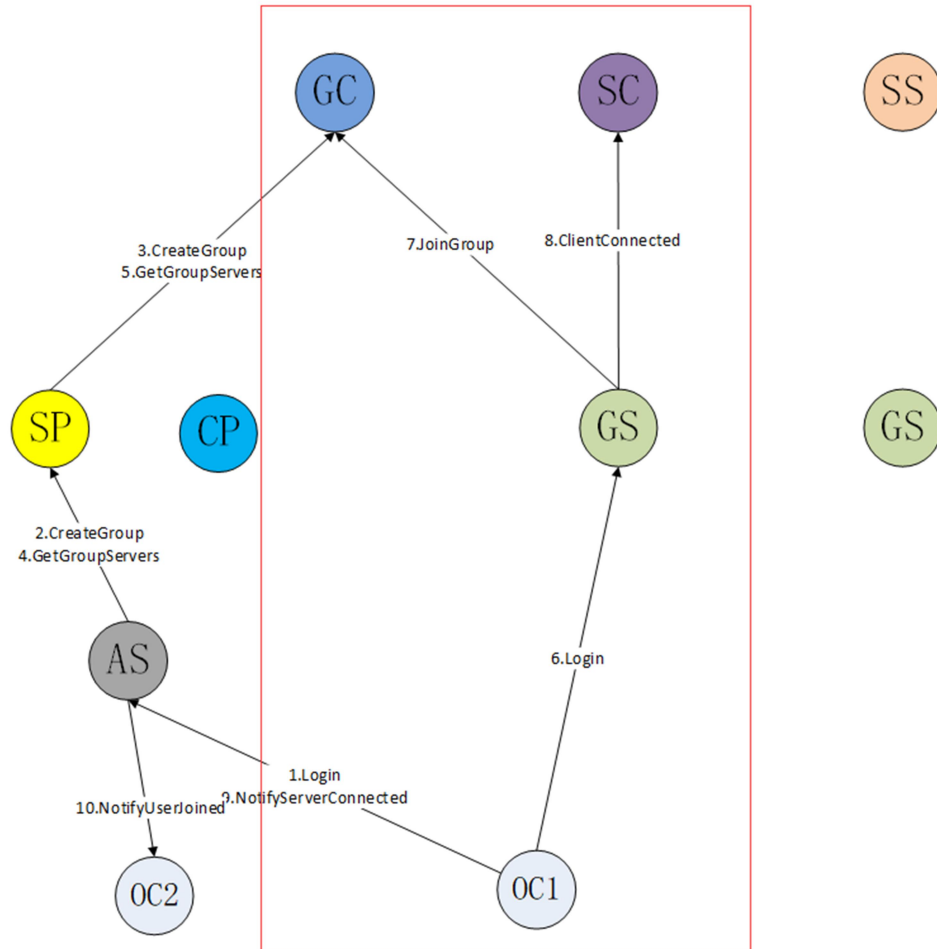
三、实现说明

讲解要点：1.客户端登录 gs 的异步实现流程，2 客户端订阅流的异步实现流程

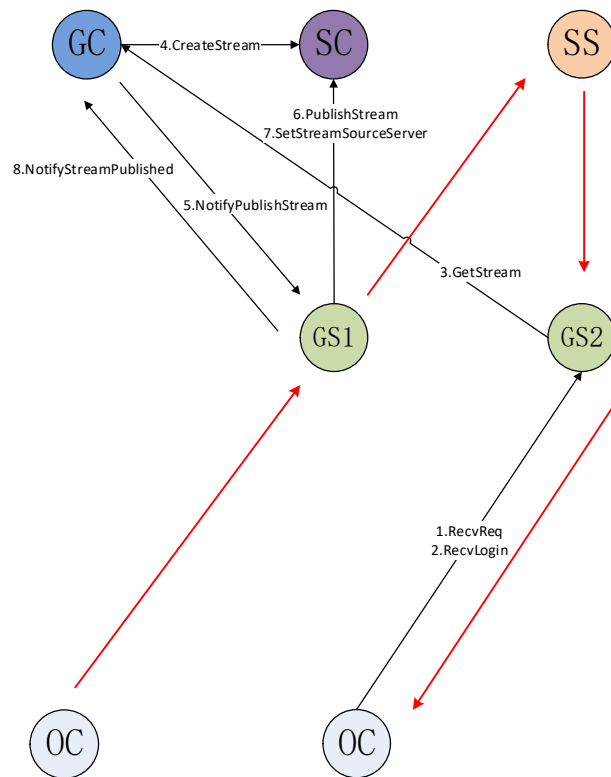
客户端在和 as 进行通讯获得当前会议的一个唯一的标识 groupId, 且拿到最佳媒体接入服务器 gs 后，向该 gs 创建信令通道会话，并登录该 gs，gs 收到客

户端登录请求后需要做两件事, 向 gc 发送 joinGroup, 向 sc 发送 clientConnected, 这两步成功后返回客户端登录成功, 客户端再向 gs 发送订阅流, 以接收源客户端广播的媒体。通讯流程如下

1、客户端登录：



2、客户端订阅流



四、代码文件

目前只提供了一个代码文件，参见代码文件 `gs.cpp`

五、总结

为了设计简单高效，我们采用了工作线程单线程异步模型，其他通讯业务线程的所有操作都向单一的工作线程抛消息，在工作线程进行数据同步处理，有效避免多线程间竞争互斥资源加锁，同时大大增加了处理请求的并发。